Simple Compiler

by using Python

DELIKANLI MERTCAN SUSUZ
ETH ZURICH

15.08.2016

0.1 Introduction

This report includes my research,study and practice for my internship project, making a simple compiler using Python programming language.

0.2 What is Compiler ?

A compiler is a computer program that transforms source code written in a programming language into another computer language, with the latter often having a binary form known as object code.

Figure 1: What is Compiler?

0.3 Components of a Compiler

A compiler is often made up of several components, one of which is a parser. A common set of components in a compiler is:
Lexer - break the program up into words.
Parser - check that the syntax of the sentences are correct.
Interpreter - explicitly execute stored precompiled codemade by a compiler which is part of the interpreter system.
Semantic Analysis - check that the sentences make sense.
Optimizer - edit the sentences for brevity.
Code generator - output something with equivalent semantic meaning using another vocabulary.

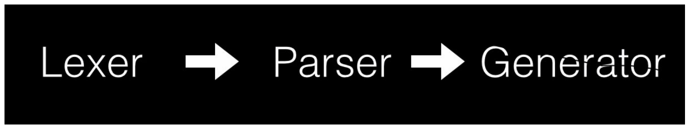
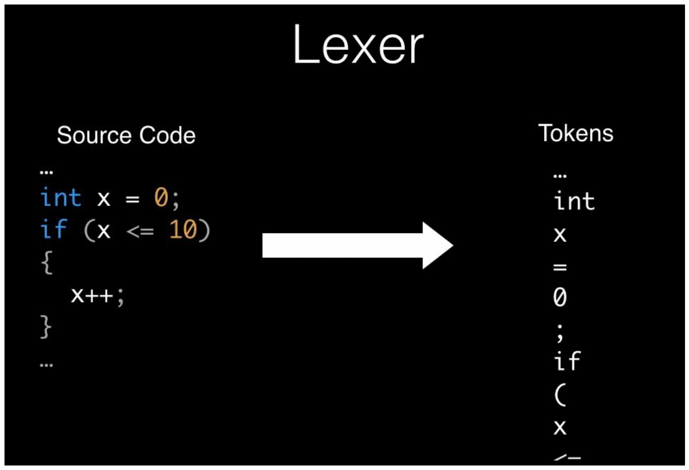


Figure 2: The goal of a compiler is to translate a source program in some high-level language into some other form.

0.4 Lexer

The process of breaking the input string into tokens is called lexical analysis. So, the first step your compiler needs to do is read the input of characters and convert it into a stream of tokens. The part of the compiler that does it is called a lexical analyzer, or lexer for short.



```
…
int x = 0;
if (x <= 10)
{
  x++;
}
…
```

```
…
int
x
=
0
;
if
(
x
```

Figure 3: Lexer

0.5 Parser

The process of recognizing a phrase in the stream of tokens is called parsing. And the part of a compiler that performs that job is called a parser. Parsing is also called syntax analysis, and the parser is also aptly called, a syntax analyzer.

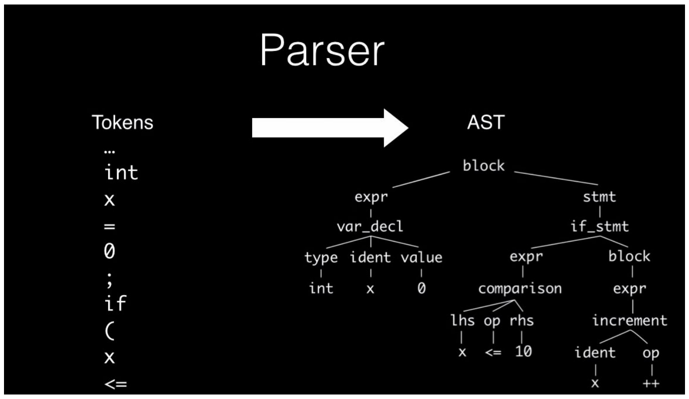


Figure 4: Parser

0.6 Trees

What is a parse tree? A parse-tree is a tree that represents the syntactic structure of a language construct according to our grammar definition. It basically shows how your parser recognized the language construct or, in other words, it shows how the start symbol of your grammar derives a certain string in the programming language.
What is abstract-syntax trees(AST)? The AST captures the essence of the input while being smaller.

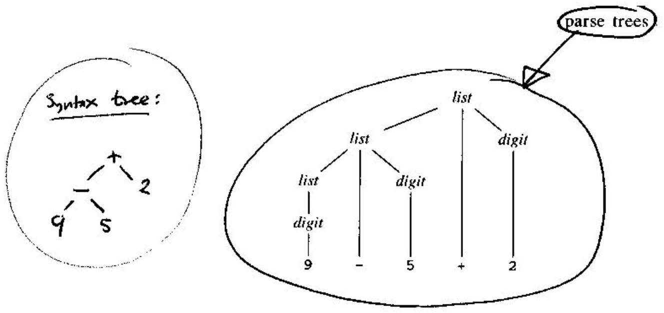


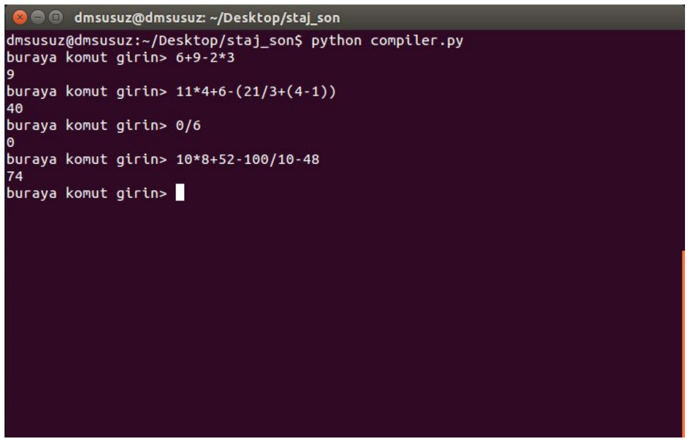
Figure 5: Parse Tree and AST for ”9-5+2”

Here are the main differences between ASTs and Parse trees:

1-ASTs uses operators/operations as root and interior nodes and it uses operands as their children.
2-ASTs do not use interior nodes to represent a grammar rule, unlike the parse tree does.
3-ASTs dont represent every detail from the real syntax (thats why theyre called abstract) - no rule nodes and no parentheses, for example.
4-ASTs are dense compared to a parse tree for the same language construct.

0.7 Analysis / Testing

I always work on terminal and tested my code multiple times.



```
dmsusuz@dmsusuz: ~/Desktop/staj_son
dmsusuz@dmsusuz:~/Desktop/staj_son$ python compiler.py
buraya komut girin> 6+9-2*3
9
buraya komut girin> 11*4+6-(21/3+(4-1))
40
buraya komut girin> 0/6
0
buraya komut girin> 10*8+52-100/10-48
74
buraya komut girin>
```

Figure 6: Terminal

0.8 The Code

I used python programmin language in this project.I learned so many new things about computer and machine learning,also learned new methods and things on python. The code attached on the mail.

0.9 Conclusion

In this process, I learned a lot of things about computer, machine learning and how the computer works. I tried to implement a simple compiler which understands binary operations. I used python programming language and also learned a lot of knowledge for python. I used some methods for the first time.

0.10 References

-Compilers: Principles, Techniques, and Tools by *Alfred V. Aho , Ravi Sethi, Jeffrey D. Ullman*
-Writing Compilers and Interpreters: A Software Engineering Approach by *Ronald Mark*
-stackoverflow.com
-coursera.org